9/29/2020

**Lab 4 Report**

Authors: Anusha Chitranshi, Abhigyan Acherjee, Athulya Ganesh

1. The objective of this lab is to examine generalization/specialisation in OOA/D/P. Concepts include: deriving a class from a base class, polymorphism, inheritance, abstract classes, forming function templates etc.
   1. The main **advantages of inheritance** are code reusability and readability. When child class inherits the properties and functionality of parent class, we need not to write the same code again in child class. This makes it easier to reuse the code, makes us write the less code and the code becomes much more readable.
   2. **Polymorphism** helps the programmer to reuse the codes, i.e., classes once written, tested and implemented can be reused as required. It saves a lot of time and a single variable can be used to store multiple data types.It also makes it easier to debug the codes.
   3. **Abstraction** allows us to create a general idea of what the problem is and how to solve it. The process instructs us to remove all specific details, and any patterns that will not help us solve our problem. This helps us form our idea of the problem.The main advantage of using a function template is to avoid unnecessary code repetition. This ultimately supports small size of byte code and compactness in the program.

It is preferable only when the statements are the same in multiple overloaded functions but just the data type on which they operate is different. This avoids multiple declaration and definition of the functions.

* 1. **Function Templates**- The main advantage of using a function template is to avoid unnecessary code repetition. This ultimately supports small size of byte code and compactness in the program. It is preferable only when the statements are same in multiple overloaded functions but just the data type on which they operate is different

2. Include in the submission ***how each member will be available*** in derived classes (i.e., not available, available if not overridden, etc.). Complete this before moving on to task 2.

The member functions in the base class Game, that is Play(),Winner(), the getter and setter functions will be public and hence available for access to the derived classes, videoGame and boardGame respectively. The constructors of the base class Game will however not be available to the derived classes, as they are inherently used to initialize objects of the base class. The private member variables of the base class will not be available to the derived classes. They remain private to the base class, and can only be accessed by the member functions of the base class.

3 . Include in the submission ***what version of the derived class members will be available*** in instances of the ***derived class*** and in instances of the ***derived class declared as the base*** class type. Complete this before moving on to task 3.

For the instances of the derived class: they can access the public member functions of the base class and the derived class as well. They however, cannot access the private members of either the base class or the derived class, or the constructors of the base class. Only the member functions of the derived class can access the private member variables of the derived class and the member functions of the base class can access its private variables.

For the instances of the derived class declared as base class: they can only access the public member functions of the base class regardless of the type of object it points to. It cannot access member functions that belong to the derived class exclusively. And it cannot access the private member variables of either base or derived class.